FILE COPY

2

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
UNLIMITED Approved for public release; distribution unlimited.

DULE

AD-A208 327

IBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 89-0609

6a. NAME OF PERFORMING ORGANIZATION
UNIVERSITY OF SOUTHERN CALIFORNIA

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
USAF, AFSC

6c. ADDRESS (City, State, and ZIP Code)
UNIVERSITY PARK
LOS ANGELES, CA. 90089-1147

7b. ADDRESS (City, State, and ZIP Code)
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
BUILDING 410
BOLLING AFB, D.C. 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL (If applicable)
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F49620-87-C-0005

8c. ADDRESS (City, State, and ZIP Code)
Bldg. 410
Bolling AFB DC 20332-6448

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61102F	2304	A2	

11. TITLE (Include Security Classification)
RESEARCH IN KNOWLEDGE DELIVERY

12. PERSONAL AUTHOR(S)
HOVY, EDUARD H.; MANN, WILLIAM C.

13a. TYPE OF REPORT
FINAL TECHNICAL

13b. TIME COVERED
FROM 11/86 TO 09/88

14. DATE OF REPORT (Year, Month, Day)
1989, MARCH, 28

15. PAGE COUNT
25

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report summarizes the research and developm nt work done over four years toward the goal of automatically planning and generating fluent multisentence paragraphs of English text. The work consisted of three principal components, namely knowledge representation, grammar development, and text structuring. With respect to knowledge representation, a powerful technique of linking the generator with arbitrary applications was developed by using a very general underlying taxonomy of entities in the world and various specific domain-related taxonomies. As part of grammar development, the invertibility of the grammar in use by the project was investigated, with the eventual goal of developing a combined bidirectional parsing-generation system using the same grammar network. Finally, a text structure planner was developed and the whole system was successfully used to generate paragraphs in three different application domains.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr. Abraham Waksman

22b. TELEPHONE (Include Area Code)
(202) 767-5027

22c. OFFICE SYMBOL
NM

Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

RESEARCH IN KNOWLEDGE DELIVERY

Final Technical Report
AFOSR Contract F49620-87-C-0005
11/1/86 - 9/30/88

Eduard H. Hovy
William C. Mann

University of Southern California
Information Sciences Institute

Abstract

This report summarizes the research and development work done over four years toward the goal of automatically planning and generating fluent multisentence paragraphs of English text. The work consisted of three principal components, namely knowledge representation, grammar development, and text structuring. With respect to knowledge representation, a powerful technique of linking the generator with arbitrary applications was developed by using a very general underlying taxonomy of entities in the world and various specific domain-related taxonomies. As part of grammar development, the invertibility of the grammar in use by the project was investigated, with the eventual goal of developing a combined bidirectional parsing-generation system using the same grammar network. Finally, a text structure planner was developed and the whole system was successfully used to generate paragraphs in three different application domains.

1 Objectives of the Research Effort

In 1985 USC/ISI proposed a four year plan for work on knowledge delivery to conduct research to enable computers to express information in fluent multiparagraph English. This is the final summary of the work conducted under this plan. The principal research themes are:

- Knowledge Representations
- Grammar Development
- Text Structure

1.1 Knowledge Representations

As in all the symbolic disciplines, development of notation is a central part of progress in the art. Text generation shares with the rest of AI a number of difficult and crucial problems with notation. The representation of information, a major subdiscipline of AI, consists almost entirely of theoretical and experimental studies of notations for knowledge.

Although many computationally useful knowledge notations exist, their collective scope is far from comprehensive, due partly to inefficiency, and partly to formal limits. Some notations are very general, such as those based on first- (and higher-) order predicate calculus and the lambda calculus, but are difficult to work with in practical systems; others are more convenient and efficient, but apply only to relatively narrow domains of knowledge. Many varieties of information which are representable in principle using existing notations often have no computationally tractable notations. There are no efficient general-purpose notations.

Researchers of language generation are in a uniquely advantageous position to investigate the adequacies and properties of various representation notation systems. This is so because English is itself organized around elementary varieties of knowledge that are highly recurrent, that are important to people, and are crucial in the solution of a great diversity of problems of everyday existence. English is a highly evolved notation, specialized over centuries of use to carry great varieties of knowledge. By developing and testing notations for linguistically prominent kinds of knowledge, researchers in language generation can help the development of solid and powerful knowledge representation systems.

During the course of the contract, we concentrated on three goals in this regard:

1. to expand existing knowledge notations to represent knowledge which is particularly crucial to generation;
2. to create new specialized notations to represent particular sorts of knowledge;
3. to develop techniques for reconciling and relating notations.

In particular, we have investigated what knowledge notations best support the delivery of knowledge in English. This work is described in Section 2.1.

1.2 Grammar Developments

The central purpose of the grammar development task was to add the functionality of language analysis (parsing) to our existing large systemic generation-oriented grammar of English. This was attempted for several reasons, the most prominent being the widespread desire among AI and Computational Linguistics workers for bidirectional grammars in which the two directions (analysis and synthesis) are compatible in both theoretical orientation and detail.

The approach taken was to reexpress the grammar, Nigel, in the Functional Unification Grammar framework, then to attempt to parse by techniques which are a variant of existing unification parsing methods. At the beginning there were several issues:

1. Formal: Could the inverse of the generation grammar be found?
2. Efficiency: Unconstrained Unification has a reputation for being exponentially slow in principle and extremely slow in practice. Would this be true of an inverted Nigel?
3. Grammatical Specificity: No Systemic grammar had ever been examined for ambiguity behavior. Would we find large factors of preventable ambiguity?
4. Inversion-specific Inefficiencies: Does analysis using a grammar designed only for generation have unsuspected deficiencies in available information?

Though this work has not been completed to the full level of generality of our grammar, a number of very promising initial results have been obtained. These are described in Section 2.2.

1.3 Text Structure

Knowledge delivery in English would remain weak and ineffective if it were restricted to using single isolated sentences. Although single-sentence generation has a place, it is important to develop an understanding of how to create larger texts. As a theoretical problem, we need to understand how texts are built up out of sentences, what special effects arise from using combinations of sentences, and how particular organizations of text should be selected or constructed. Until recently, text generation research has been hampered by a lack of suitable descriptive theory; existing descriptions have been to be too informal and too literary to be computationally useful.

The heart of the problem is that of text coherence. Coherent text can be defined as text in which the hearer knows how each part of the text relates to the whole; i.e., (a) the hearer knows why it is said, and (b) the hearer can relate the semantics of each part to a single overarching framework.

The problem of text coherence can be characterized in specific terms as follows. Assuming that input elements are sentence- or clause-sized chunks of representation, the permutation set of the input elements defines the space of possible paragraphs. A simplistic, brute-force way to achieve coherent text would be to search this space and pick out the coherent paragraphs. This search would be factorially expensive. For example, in a paragraph of 7 input clusters, there would be $7! = 5{,}040$ candidate paragraphs. However, by utilizing the constraints imposed by coherence, one can formulate operators that guide the search and significantly limit the search to a manageable size.

The force of our research is that, exercising proper care, the coherence relations that hold between successive pieces of text can be formulated as the abovementioned search operators and used in a hierarchical-expansion planner to limit the search and to produce structures describing the coherent paragraphs. The state of this research is described in Section 2.3.

2 Status of the Research Effort

2.1 Knowledge Representation

To guide our work, we used the following criteria of importance and readiness:

1. Prefer varieties of knowledge that are required in every sentence over those that are optional.

2. Prefer varieties of knowledge whose representation will support other subtasks of this and related research.

3. Prefer varieties of knowledge for which the corresponding parts of our particular grammar are well elaborated.

4. Prefer varieties of knowledge for which we and/or others have already developed attractive proposals on how to represent the knowledge.

Based on these criteria, we concentrated on developing the representation of actions, their participants, and propositional relations.

As our basic knowledge representation system, we used two descendants of the KL-ONE knowledge representation formalism, namely NIKL and LOOM. Both NIKL (New Implementation of KL-ONE) and LOOM, a successor to NIKL, were designed and built at ISI. LOOM, which is still under construction, has many extensions over NIKL, which in turn had some desirable properties beyond those of KL-ONE. (The classifier, a mechanism which automatically classifies a newly-defined entity in terms of the existing definitions based on the aspects and properties of the entity, is a very useful feature.)

Our primary uses of the NIKL (and now, LOOM) systems are to represent the information to be expressed. This information is typically produced by some application system, such as an expert system or a data base access program, which has to communicate its output needs to the generator using a language of common terms and structure.

The inputs given to a generator must be intelligible to it. Therefore, they must either be generator-internal symbols, or they must be defined in terms of its taxonomy of the entities of the world. Our generator Penman uses a taxonomy of the types of entities that appear in the world because different types of entities are said differently in English (for example, actions are typically expressed as verbs and objects as nouns). Without such a taxonomy, Penman would have no way of determining whether to treat a symbol as an object, as an action, as a relation, etc. The categories defined in the Upper Model reflect grammatical distinctions made in English.

This taxonomy consists of two parts, called the *Upper Model* and the *Domain Model*. The Upper Model entities create a very abstract partitioning of the world, and the Domain Model contains subordinate entities that create increasingly fine, more everyday, task-oriented distinctions. The Upper and Domain Models constitute a generalization hierarchy organized as a property-inheritance network. When a new symbol is defined, it is placed in the taxonomy relative to one or more existing symbols, from whom it inherits features in addition to the particular features it is defined to have.

The Upper Model: The top node of the Upper Model is the entity THING. The next level contains three subclasses, OBJECT, QUALITY, and PROCESS (which respectively organize objects (such as "ship"), qualities (such as "red", "operational"), and processes. The category PROCESS is divided into four types: MENTAL-PROCESS, VERBAL-PROCESS, MATERIAL-PROCESS, and RELATIONAL-PROCESS. Mental processes are such actions and states as "think" and "believe"; verbal processes are such actions as "tell" and "read"; material processes are the remaining actions and events, such as "sail" and "eat"; relational processes represent static relations, such as ownership, times, locations, etc.

By virtue of their positions in the inheritance hierarchy, entities inherit aspects or roles from their ancestors. Some of the commonly used aspects are:

- **domain:** any 2-place relation that is defined in the Upper or Domain Models holds between a domain and a range. This is the generic first place of a relation.
- **range:** The generic second place of a relation.
- **actor:** expresses the doer — the agent — of any MATERIAL-PROCESS, that is, of any action or event.
- **actee:** expresses the direct patient of any MATERIAL-PROCESS.
- **class-ascription:** expresses class membership (that is, the basic IS-A or A-KIND-OF subsumption relation).
- **property-ascription:** a general relation to express some property of an object.

The Upper Model is distinctive in that it reflects general category distinctions found in language in a way that most organizations of knowledge do not. For example, qualities are distinguished from classes (comparable to the adjective/noun distinction) rather than simply being treated alike as predicates. The Upper Model currently contains about 180 entities. With it, our system spans a large subset of English; in the three domains of experimentation we have tried it, we have not had to expand the Upper Model to any large extent.

The Domain Model: The Domain Model contains the definitions of the entities particular to the current application domain. This model should constitute a full ontology for the domain, defining all the types of objects, actions, relations, states, etc., that are used. Most applications require such a model as a natural part of their work, either explicitly or implicitly (for example, the field types and the relations among them in relational data bases).

These entity definitions in the domain model must be subordinated to entities in the Upper Model. That is to say, the entities defined in the Domain Model must form a hierarchy that can be knitted to the Upper Model in such a way that the inheritance proceeds smoothly down from Upper Model entities to increasingly specific Domain Model entities. Subordination provides the generation system with the general type of each domain entity used. In addition, subordination provides the inheritance of the aspects (roles) that domain entities' ancestors take, as well as the accompanying constraints (number constraints, filler requirements, etc.). For example, if the entity CAR is subordinated to the entity VEHICLE, and VEHICLE is defined with the aspect AGE whose filler requirement is NUMBER, then a CAR will inherit the requirement that it have a numerical age. If the generator is then able to express the AGE aspect for one entity, it can express it for all entities. When a new entity is defined — say, FORD — and subordinated to CAR, then it inherits the AGE aspect and its filler requirement.

This inheritance of aspects and requirements is very useful. All the basic aspects of actions and relations have been defined in the Upper Model, which means that the user of the system has little additional work to do. Since every entity in the application domain must have (an) Upper Model ancestor(s), every domain entity will inherit a set of aspects from the Upper Model. (Of course, the entity may have additional domain-specific aspects as well.) This is one example of the power gained by a felicitous choice and use of knowledge representation system.

These accomplishments serve our needs on action and participant relations particularly well, because they test action- and participant-oriented notations in both relatively language-neutral and relatively language-intense contexts. Recent project activity has involved coordinating these notations with other notions strongly related to action, such as events, times, places, outcomes, products and beneficiaries. They also serve well to test our ideas about propositional relations. Clause coordination in English and propositional relations in knowledge notation are in some ways two sides of the same coin. We have already been able to demonstrate relative clauses in English, along with English expression of time relations; we have also demonstrated several varieties of clause coordination. (Our generator's grammar currently allows 59 different kinds of clause combination, including 16 varieties of relative clause.)

It is important to note that the knowledge representation problem here is not a problem of whether the notations will in principle provide expressibility of particular information. Rather it is a problem of providing usable, manageable, compatible techniques for expressing a diversity of information.

The subsections below describe our approach to developing useful notations for particular varieties of knowledge.

2.1.1 Knowledge of Actions and Participants

English and other languages have elaborate provisions for describing actions and their participants. Of the two principal sentence types (Relational and Material), one is organized around an action expressed in the main verb, usually with other parts of the sentence devoted to identifying the participants in the action, such as its agent and the objects acted upon. To be able generate texts, it is important to have control of the grammar of actions, and equally important to be able to represent efficiently the knowledge of actions.

AI's weakness in action representation is well recognized. One style of knowledge representation, the so-called frame oriented languages, are relatively well suited for the representation of action. However, it typically shows no strong differentiation between participant identification and other knowledge, and does not treat actions in a way that distinguishes them notationally from objects, states, relations, or other entities. The organization of natural languages suggests that there is a strong advantage to making such distinctions highly accessible. For example, many English words represent things from the point of view of participants in actions — words like "pilot", "researcher", and "observer"; there are also specialized suffixes used only to indicate participant roles: grant*or* and grant*ee*. These enable communication and inference about actions, such as granting, independent of possible type distinctions (e.g., persons vs. institutions) among various participants. Rather than develop supplementary notations, we have extended an existing frame-oriented notation to provide more specifically for actions and their participants.

The use of the Upper Model is based on a strategy in which grammatical decisions are converted into taxonomic discriminations. Experience with this approach has been successful, but has also identified some problems. One of these problems arises because taxonomic distinctions derive from two sources: the linguistic conventions of English and the knowledge representation conventions of the host system for which sentences are generated. For example, a data base about travel may represent several kinds of trips: long and short trips, convention and conference trips, sales and recruiting trips. All of these may be represented in the data base as undifferentiated attributes of trips. Linguistically, *long* and *short* are attributes, best represented in the upper structure as qualities. Conventions and conferences are best represented as events, and sales and recruiting are best represented as kinds of processes or activities. Knowing these distinctions is essential to making the grammatical choices involved in talking about them.

The difficulty is that the regularity and homogeneity of the host system's knowledge needs to be retained, to keep it well organized and maintainable, but at the same time the linguistic differences need to be represented taxonomically for language generation. We are currently exploring several proposed solutions to this problem, but it has not yet been solved.

2.1.2 Knowledge of Propositional Relations

The expressive resources of English devoted to actions are strongly related to those devoted to propositions — roughly the expression of notions which take truth values. These resources are rich, including many methods for relating one proposition or action to another. The conjunctions and subordinators (including "and", "but", "when", "if", "although", "for instance", "that is", "so", "because", "then", "until", "while", and many more) are one part of this resource.

The weakness of AI notations in this area is well known. Notations oriented toward logic often do well with *and* and *or*, but the formal notation departs strongly from ordinary English usage. The other terms are more problematic. More diversity appears in the corresponding parts of frame-oriented notations, but there is relatively little language-oriented experience.

We approached this problem as follows:

1. In the Upper Model mentioned above, relations are given a distinguished place.
2. Within the relations subhierarchy, relations between propositions are given a distinguished place, and are further subdivided.
3. A small number of expressive facilities of the grammar are programmed to recognize particular interpropositional subtypes and employ the corresponding special facilities of English to express them.

The general strategy is to recognize in the high level knowledge organization conceptual distinctions that are important in English expression, and to use those distinctions in delivering the knowledge.

2.2 Grammar Developments

Though this work has not been completed, significant enough progress has been made to provide answers for some of the questions listed above, namely,

1. Formal: Could the inverse of the generation grammar be found?

2. Efficiency: Unconstrained Unification has a reputation for being exponentially slow in principle and extremely slow in practice. Would this be true of an inverted grammar?

3. Grammatical Specificity: No Systemic grammar had ever been examined for ambiguity behavior. Would we find large factors of preventable ambiguity?

4. Inversion-specific Inefficiencies: Does analysis using a grammar designed only for generation have unsuspected deficiencies in available information?

A manageable formal inverse has been found and is being used. A principal step in finding the inverse was the Ph.D. work of project member Robert Kasper, and this finding was reported in [Kasper 87a, 87b]. Although the brute-force approach would be extremely inefficient, ways have been found to make the system's speed reasonably acceptable for experimental use. Fortunately, the framework did not exhibit the feared large ambiguity factors [Kasper 88]. This is partly because the relatively semantic orientation of the grammar provides some additional non-traditional selectivity. However, many of the classical sources of ambiguity still occur, since there is a significant amount of inherent ambiguity in every natural language. With respect to the grammar, several varieties of inversion-specific inefficiencies were found and corrected. We now believe that a bidirectional systemic grammar must have a few small parts dedicated solely to generation and to analysis, but that nearly all of its parts can be shared. Thus the efforts of extending and maintaining this sort of bidirectional grammar are about the same as for a single-directional grammar.

There is growing interest in the research community in bidirectional grammars. Recent work on systems containing two unidirectional grammars has shown the difficulty of maintaining compatibility. A general consensus is developing that there is now enough knowledge of how to make a grammar bidirectional that future systems should have a single grammar for both directions.

2.3 Text Structure

The earliest feasible computational approach to the problem of producing coherent multi-sentence paragraphs involved the use of paragraph-sized structures called schemas which were essentially templates into which sentences could be fitted [McKeown 82]. Beginning in 1983 we developing a more suitable body of theory, so that there is now a strong descriptive foundation to build on, which we have called *Rhetorical Structure Theory* (RST) [Mann & Thompson 83, 87a, 87b, 88a, 88b, Mann [illegible] 88b, 88c, Mann, Matthiessen & Thompson 88, Thompson & Mann 86, 87]. In RST, texts are described by rhetorical schemas, each of which relates several spans of text. Schemas are defined in terms of relations between a focal span, called the nucleus, and satellite spans. For example, one schema describes a span consisting of two smaller spans, one of which identifies a problem and the other of which identifies a solution to that problem. This schema is called *solutionhood*. Another schema, *evidence*, describes the combination of a claim and evidence for it. About

25 schemas have been identified, after a study of over 200 texts, spanning scientific article abstracts, cookbooks, letters, magazine articles, and so forth.

RST as a whole is recursive: its schemas describe the individual spans of text which are described in combination by other schemas. As a result, it is capable of describing a wide range of sizes of text unit, an unusual property among descriptions of text structure. It accounts for coherence properties, certain facts about text order, and many observations about conjunctions and relations between propositions.

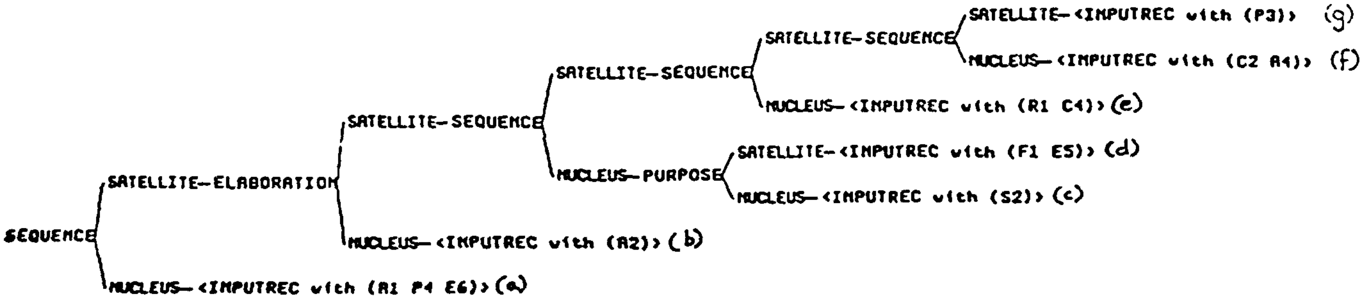
In 1987 a major new definitional basis for RST was completed [Mann & Thompson 87a]. It is more explicit than previous work, and grounded in direct study of the RST of a larger number of actual texts. This work coincided with the first computational implementation of RST [Hovy 88a, 88b]. In the implementation, the definitions of RST schemas identified recognition criteria with notions of the speakers' goals. This provides a basis for relating the coherence of text, as governed by RST relations, to the speaker's intentions in producing the text. Thus RST is essentially a goal-based theory. Its descriptions are organized around the intentions of the speaker and the part-to-part relations in the text which are used to carry out those intentions. In this it is comparable to recent work by Grosz and Sidner, but it does not work exclusively with the kind of fine-grain axiomatization of intention which they hope for [Grosz & Sidner 86].

Penman's text structure planner operates in top-down hierarchic expansion fashion, patterned after the planning system NOAH [Sacerdoti 77]. In the example shown in Figure 1, an expert system that suggests changes to computer code to improve its readability and maintainability, provides the planner with a collection of 7 units of information, gathered from its procedural knowledge, as well as the goal to explain the reasoning behind its recommendations. Using this goal to start planning, the text planner uses its library of RST relation/plans to build a tree in which branch points are RST relation/plans and leaves are input elements. It then traverses the tree, sending the leaves' content to the generator to be transformed into English. The tree in figure 1 gives rise to the paragraph shown below the tree. It contains the RST relations SEQUENCE (signalled by "then" and "finally" in the paragraph), ELABORATION ("in particular"), and PURPOSE ("in order to").

The operationalization of the RST relations as plans is currently incomplete, both in the number of relations handled and in the combination of relation/plans with schemas for enhanced planning capability. We have operationalized only six of the twenty most basic RST relations. Operationalization involves formalizing the restrictions on a relation's use and the requirements for its parts in a language built from the formal theory of rational interaction currently being developed by, among others, Cohen, Levesque, and Perrault. For example, in [Cohen & Levesque 85], Cohen and Levesque present a demonstration that under certain assumptions the indirect speech act of requesting can be derived (recognized) using the following basic modal operators

- **(BEL x p)** — p follows from x's beliefs
- **(BMB x y p)** — p follows from x's beliefs about what x and y mutually believe

Figure 1: Paragraph structure tree for PEA text.


SATELLITE—<INPUTREC with (P3)> (g)
SATELLITE—SEQUENCE
NUCLEUS—<INPUTREC with (C2 A4)> (f)
SATELLITE—SEQUENCE
NUCLEUS—<INPUTREC with (R1 C4)> (e)
SATELLITE—SEQUENCE
SATELLITE—<INPUTREC with (F1 E5)> (d)
NUCLEUS—PURPOSE
NUCLEUS—<INPUTREC with (S2)> (c)
SATELLITE—ELABORATION
SEQUENCE
NUCLEUS—<INPUTREC with (A2)> (b)
NUCLEUS—<INPUTREC with (A1 P4 E6)> (a)

[*The system asks the user to tell it the characteristic of the program to be enhanced.*]$_{(a)}$ **Then** [*the system applies transformations to the program.*]$_{(b)}$ **In particular,** [*the system scans the program*]$_{(c)}$ **in order to** [*find opportunities to apply transformations to the program.*]$_{(d)}$ **Then** [*the system resolves conflicts.*]$_{(e)}$ [*It confirms the enhancement with the user.*]$_{(f)}$ **Finally,** [*it performs the enhancement.*]$_{(g)}$

- **(GOAL x p)** — **p** follows from x's goals
- **(AFTER a p)** — **p** is true in all courses of events after action **a**

We are using these relation/plans as compilations of these operators and the logical operations **AND** and **OR**. The operationalization task is difficult because one must ensure that the restrictions and requirements are formalized in ways that are at once specific enough to be directly useful in a computer program while being general enough to be applicable to the wide range of purposes for which the relations were originally intended.

The relation/plans we have formalized thus far — SEQUENCE, ELABORATION, PURPOSE, etc. — have enabled us to produce a number of paragraphs in three different domains of application (discussed in the next section). However, these six relations have not been sufficient to produce all the kinds of texts one could produce from these domains. Even so, this method of planning coherent paragraphs has already aroused interest in the Natural Language Processing community. In addition, we have discovered that it is possible to formulate these plans as schemas, or even to form hybrids that are a mixture of schemas and plans. This finding is very encouraging, because it makes available applications for paragraph structuring that otherwise are very difficult or impossible to perform. That is to say, relation/plans are useful primarily when a large amount of flexibility is desired over a relatively small number (in the order of 10 to 30) of clause-sized units of information to be conveyed. However, in large collections of information, a less flexible method with more structure is required, if planning time is to be kept manageable — and this is exactly the functionality of schemas. As explained in [Hovy 88b], it is possible to treat the growth points in relation/plans — those points that suggest the inclusion of additional material to the planning process — either as suggestions (in which case you get flexible planning) or as injunctions (in which case you get schemas). This finding has not been extensively tested yet, and we have not integrated this notion into the planning system. We will explore this avenue of research, developing criteria for deciding when to follow the flexible planning route and when to follow the schema route. The hybrid approach combines ease of implementation and tight control with flexible, need-driven planning, overcoming the shortcomings of either technique taken alone.

2.4 Current Collaborations

There is a methodological problem in developing knowledge delivery techniques: the techniques must somehow be tested and refined so that they work. Proofs of sufficiency-in-principle are not enough. The complexity of the subject makes it necessary to develop techniques on particular subject matter and knowledge rather than always working directly on the general case.

To meet this need we have begun to create a series of experimental text generation systems that embody the notations and processes being studied. The first of our series of experimental systems contained knowledge about computer versions of personal mail and appointment calendars. It was developed in conjunction with a related DARPA project which attempted to apply existing state-of-the-art technology to the problem of interacting with data bases in English. The DARPA project, part of the Strategic Computing program, served as a testbed for many of the ideas from Knowledge Delivery Research, and made it much easier to refine and extend these ideas.

With the development of an implementation of a planner that used some relations of RST, we had limited multisentential capability about two years ago. The mail and calendar system was replaced by collaboration with the following three projects (funded separately):

1. An integrated multimedia interface system (II), in which paragraphs of English text, planned and generated by Penman, are combined with maps, menus and other display methods, so as to be suitable for command and control use. As part of this work, a naval briefing environment was captured in which the English presented information derived directly from a (sanitized) US Navy assets database. The project team was led by Dr. Norman Sondheimer.

2. The Program Enhancement Advisor (PEA) is an experimental expert system that interactively advises programmers on how their Lisp programs might be improved. It contains an explanation facility that uses Penman's grammar to generate text that explains how PEA works. PEA is being developed as a Ph.D. project by Johanna Moore under the direction of Dr. Bill Swartout.

3. The Digital Circuit Diagnosis system (DCD) is an experimental expert system that diagnoses faults in digital hardware. Like PEA, it contains an explanation facility that uses Penman's grammar to generate output. Text is generated that explains the definitions of entities within DCD and the reasoning that lead to the diagnosis. DCD is being developed by Dr. Cécile Paris in collaboration with Dr. Bill Swartout.

2.4.1 Application to Briefings from a Military Data Base

The first test of the multisentence planning capability was performed on data provided by the Integrated Interfaces application domain. In response to a user's request for information from a data base of Naval deployments, the II system gathered appropriate information and distributed it to its various output modes, one of which was the text planner and Penman. Some sample paragraphs generated by Penman in this domain were:

```
Knox, which is C4, is en route to San Diego in order to
rendezvous with Task Group CTG70.1. It will arrive 4/24.
It will perform exercises for four days.

Kennedy and Merrill are on a multisail to Sasebo, arriving
10/19. While it is in Sasebo, Kennedy, which is C4, will
load until 10/22. Merrill will depart on 10/20 to be on
```

```
operations until 10/30.

MEKAR-87 takes place in South China Sea from 10/20 until
11/13. Preble, Fanning, and Whipple are participating.
Preble and Fanning arrive 10/20. Whipple arrives on 10/29.
Preble, which is C3, will leave on 10/31. Fanning and
Whipple will leave on 11/13.
```

2.4.2 Application to a Program Enhancement Advisory Tool

Beginning late in 1987, Penman was interfaced to the Program Enhancement Advisor (PEA), part of an independent research project at ISI. This step was particularly significant because PEA is a member of a design family of systems that are specially organized for knowledge delivery. It is built in the Explainable Expert Systems framework, a generalization of foundational work by Dr. Bill Swartout.

It is commonly acknowledged that expert systems should be able to explain their behavior and methods, but most actual expert systems do so poorly, if at all. In the EES framework, programs are developed from the very beginning with explanation in mind, and much of the design information for a program is retained within it for use in explanations. After some initial use of our text structure planner, the PEA and DCD project members built their own text planner in roughly the same mold, affording them greater freedom of experiment, but continued to use the sentence generator Penman.

Two texts from the PEA domain, describing the expert system's internal rules and process representations are:

```
A transformation that enhances the readability of the
program is defined as a transformation whose right hand
side is more readable than its left hand side.  One kind
of a transformation whose right hand side is more readable
than its left hand side is a transformation that has a
right hand side that is a function that has a function
name that is a common English word and a left hand side
that is a function that has a function name that is a
technical word.  CAR-TO-FIRST is a transformation that
has a right hand side that is a function that has a
function name that is a common English word and a left
hand side that is a function that has a function name
that is a technical word.

The system asks the user to tell it the characteristic
of the program to be enhanced. Then the system applies
transformations to the program. In particular, the system
```

```
scans the program in order to find opportunities to apply
transformations to the program. Then the system resolves
conflicts. It confirms the enhancement with the user.
Finally, it performs the enhancement.
```

2.4.3 Application to a Digital Circuit Diagnosis System

In April, 1988 Penman was interfaced to a second program in the EES family, the Digital Circuit Diagnosis system (DCD), being developed by Dr. Cécile Paris. The DCD texts generated so far are definitional, and thus rely on different expressive facilities than PEA does.

Some interesting research has been conducted by Drs. Bateman (from the Penman project) and Paris (from the DCD project) on the generation of different surface forms of the same underlying propositional content. For example, the following three texts from the same underlying knowledge structure in the DCD domain, tailored to readers of various levels of sophistication, were generated by Penman:

```
The system is faulty, if there exists an O in the set
of output terminals of the system such that the expected
value of O does not equal the actual value of the signal
part of O and for all I in the set of input terminals of
the system, the expected value of the signal part of I
equals the actual value of the signal part of I.

The system is faulty, if all of the expected values of
its input terminals equal their actual values and the
expected value of one of its output terminals does not
equal its actual value.

The system is faulty, if the inputs are fine and the
output is wrong.
```

The work of interfacing Penman to DCD was closely monitored so that we could understand the interfacing process. This led to a report which showed that interfacing currently takes about three person-weeks of effort, eventually reducible to about one person-week. Out of the experience of these two applications, and also to overcome some of Penman's internal notational problems, a new sentence specification language called SPL (Sentence Plan Language) was developed, to serve both as an internal notation between Penman's text planner and its sentence generator, and also as an external interface language for sentence generation. SPL has greatly reduced the amount of time it takes an outside user to learn to use Penman.

In addition to collaborating with other projects within ISI, the Penman project is committed to getting Penman out to the community, both in order to have it used and tested, and in order to have other people work on extending the grammar. Currently, the Penman system runs in Common Lisp on Symbolics and TI Explorer Lisp machines. To widen its distribution, work is currently underway to port the system to the Mac-II computer (which will greatly expand its accessibility to graduate students and linguists) as well as to Sun workstations (whose window system, X, is fast becoming an industry standard). To acquire the system, a potential user must simply sign the licensing agreement, pay a nominal administrative fee, and load two tapes onto his or her Lisp machine. The loading and installation process takes less than two hours.

2.5 New Opportunities in Knowledge Delivery

The success of the Penman system in planning and synthesizing texts opens up technical possibilities that were not previously available. In addition to knowledge delivery by means of synthesis of written English monologues, there are related communicative processes which might depend primarily on the same kinds of knowledge. These include synthesis of spoken English output, communication within interactive dialogue (especially in online human-computer interfaces) and various radical revisions of the underlying technology.

Each such change involves technical constraints on the methods used to achieve communication. Many of these constraints are unknown, so it is not clear what communicative possibilities are currently feasible.

It is now timely to explore some of these possibilities. We have identified several below. For each of these we expect to devote a small amount of effort to investigating the technical feasibility of extending present and forthcoming work in the given direction.

- **Speech Synthesis from Meanings:** The current capability for written text synthesis from meanings actually produces, as a by-product, much of the information that is needed for speech synthesis.

- **Dialogue and Interface Participation:** Engaging in dialogue or English-language human-computer interaction involves keeping track of a richer diversity of information about the other participant, and also a richer notion of communication planning, than monologue requires.

- **Multiple Perspectives:** One of the limitations of the techniques embodied in Penman is that there is a single fixed point of view toward each object in the system's knowledge. The view is selected at implementation time. This makes it difficult to use grammatical options such as nominalization, e.g. to use the verb "synthesize" pr the noun "synthesis" when referring to the same process, or to use "those cows" instead of "that herd" to refer to the same group. Knowledge representation techniques that overcome this limitation are needed.

- **Alternative Control Structures:** Text generation is a complex problem involving a wide diversity of knowledge sources. Penman's control structure is a simple pipeline that attempts to anticipate all of the combinations that are important and likely. More effective control structures based on blackboards, unification, object-oriented programming, opportunistic inference and other techniques should be considered, especially for implementation of the remodularized system.

2.6 Summaries of the Principal Research Components

2.6.1 Knowledge Representation

Penman's Upper Model implements a taxonomic strategy for representing the linguistic expressive possibilities for specific kinds of knowledge. The strategy seems generally successful, but ongoing experimentation with this structure is needed to determine whether the strategy will work on very large or diverse collections of knowledge, and whether it will work when there is another organization imposed on the same body of knowledge.

The taxonomic strategy is being extended to a wide range of propositional relations, partly derived from RST, in order to test its effectiveness in a different way.

2.6.2 Text Structure

Work on constructing texts must rest on a strong descriptive theory. We now have such a descriptive theory, RST, in place, and it is being accepted by many linguists as a significant advance over what was previously available. The partial implementation of RST is useful in providing a model of how the descriptive theory can be made constructive, but the texts created so far are not big enough, diverse enough or numerous enough to judge the success of the implemented theory. These limitations can be overcome only with substantial effort in constructing experimental bodies of knowledge which are rich enough so that several interesting texts can be constructed for a given purpose. In addition, there must be attention to non-structural aspects of text planning in addition to the RST-related aspects, so that the quality of generated texts can be suitably evaluated.

These needs for extension and testing, for both knowledge representation and text structure, will be central research activities for the project in future research.

2.6.3 Grammar Development

The testing of the bidirectional grammar as an analyzer is not yet complete. Its speed is acceptable for research, and the amount of modification to the grammar needed in order to make it bidirectional is quite small. Ambiguity factors are not very different from those of grammars that yield no functional information from the analysis. Important new information has been gained (and presented publicly) about the sources of ambiguity in this class of grammars, and about how spurious ambiguity can be avoided.

Further work will make evaluation possible, and we expect to seek further research benefits after that evaluation is complete. Those benefits will certainly include application of the grammar in ISI's ongoing and future natural language understanding work, and also use of the grammar as a bidirectional research tool by outside research groups.

3 List of Publications

The following publications were written about the work sponsored under this contract:

1. Bateman, J., Kasper, R., Steiner, E. and Schütz, J. Interfacing an English Text Generator with a German MT Analyzer. In *Proceedings of the Annual Meeting of the GLDV*, Springer-Verlag, 1989.

2. Bateman, J., Kasper, R., Schütz, J. and Steiner, E. A New View on the Process of Translation. To be presented at the Conference of the European Association for Computational Linguistics, Manchester, England, April, 1989.

3. Bateman, J. Conversation Generation — a theoretical watershed? In *New Developments in Systemic Linguistics*, Fawcett, R. and Young, D. (eds.), Volume 2, Frances Pinter (to appear).

4. Hovy, E. Planning Coherent Multisentential Text. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June, 1988. Also available as USC/Information Sciences Institute Reprint RS-88-208.

5. Hovy, E. Approaches to the Planning of Coherent Text. In *Proceedings of the 4th International Generation Workshop*, Los Angeles, CA, July, 1988.

6. Hovy, E. On the Study of Text Planning and Realization. In *Proceedings of the AAAI Workshop on Text Planning and Generation*, AAAI, St. Paul, MN, August, 1988.

7. Kasper, R. A Unification Method for Disjunctive Feature Descriptions. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, Palo Alto, CA, July, 1987. Also available as USC/Information Sciences Institute Reprint RS-87-187.

8. Kasper, R. Conditional Descriptions in Functional Unification Grammar. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June 1988. Also available as USC/Information Sciences Institute Research Report RR-87-191, November, 1987.

9. Kasper, R. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary, August 1988. Also available as USC/Information Sciences Institute Reprint RS-88-212.

10. Kasper, R. Systemic Grammar and Functional Unification Grammar. In *Systemic Functional Approaches to Discourse*, Benson, J. and Greaves, W. (eds), Norwood, NJ: Ablex (in press). Also available as USC/Information Sciences Institute Reprint RS-87-189.

11. Mann, W. and Thompson, S. Rhetorical Structure Theory: A Framework for the Analysis of Texts. In *IPRA Papers in Pragmatics*, Volume 1, 1987. Also available as USC/Information Sciences Institute report RS-87-185.

12. Mann, W. and Thompson, S. Rhetorical Structure Theory: Description and Construction of Text Structures. In *Natural Language Generation: New results in Artificial Intelligence, Psychology and Linguistics*, Kempen, G. (ed.), Nijhoff, 1987. Also available as USC/Information Sciences Institute report RS-87-174.

13. Mann, W. Text Generation: The Problem of Text Structure. In *Natural Language Generation Systems*, McDonald, D. and Bolc, L. (eds), Springer-Verlag: New York, 1988.

14. Mann, W. Dialogue Games. In *Argumentation*, 1988. Also available as USC/Information Sciences Institute report RR-79-77.

15. Mann, W. Two Theories of Discourse Structure. In *Proceedings of the 4th International Generation Workshop*, Los Angeles, CA, July, 1988.

16. Mann, W., Matthiessen, C. and Thompson, S. Rhetorical Structure Theory and Text Analysis. In *Discourse Description: Diverse Analyses of a Fund Raising Text*, Mann, W. and Thompson, S. (eds), (to appear).

17. Mann, W. and Thompson, S. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. In *Text*, Vol. 8:3, 1988.

18. Mann, W. and Thompson, S. Rhetorical Structure Theory: A Theory of Text Organization. In *The Structure of Discourse*, Polanyi, L. (ed), Ablex: Norwood, NJ, 1988.

19. Matthiessen, C. Lexical Selection in Generation: An Abstract Model. In *Proceedings of the 4th International Generation Workshop*, Los Angeles, CA, July, 1988.

20. Matthiessen, C. Representational Issues in Systemic Functional Grammar. In *Systemic Functional Approaches to Discourse*, Benson, J. and Greaves, W., Ablex, 1988. Also available as USC/Information Sciences Institute report RS-87-179.

21. Matthiessen, C. and Thompson, S. The Structure of Discourse and Subordination. To appear in *Clause Combining*, Haiman, J. and Thompson, S. (eds), Amsterdam: John Benjamins, 1988.

22. Matthiessen, C. Representational Issues in Systemic Functional Grammar. In *Systemic Functional Approaches to Discourse*, Benson, J. and Greaves W. (eds), Norwood, NJ: Ablex (in press).

23. Soudheimer, N., Cumming, S. and Albano, R. How to Realize a Concept: Lexical Selection and the Conceptual Network in Text Generation. In *Proceeding of the Workshop on Theoretical and Computational Issues in Lexical Semantics.* Waltham, Massachussets, April, 1988.

24. Sondheimer, N. Lexical Selection. In *Proceedings of the Workshop on Conceptual Networks*, AAAI, St. Paul, MN, August, 1988.

25. Thompson, S. and Mann, W. A Discourse View of Concession in Written English. In *Proceedings of the Second Annual Meeting of the Pacific Linguistics Conference*, DeLancey, S, and Tomlin, R., 1986.

26. Thompson, S. and Mann, W. Antithesis: A Study in Clause Combining and Discourse Structure. In *Language Topics: Essays in Honour of M.A.K. Halliday*, Steele, R. and Threadgold, T. (eds.), Benjamins, 1987. Also available as USC/Information Sciences Institute report RS-87-171.

27. Whitney, R. *Semantic Transformations for Natural Language Production.* USC/Information Sciences Institute report RR-88-192, March, 1988.

4 Personnel

The following personnel were supported in full or in part in the duration of this contract (degrees listed were attained under partial sponsorship of this contract; recipients were either part-time project members before graduation or joined the project full-time afterward):

- Mr. Robert N. Albano (currently project member)
- Dr. John A. Bateman (currently project member)
- Ms. Susanna Cumming (currently a Linguistics Department faculty member at the University of Colorado; attained Ph.D. in Linguistics from UCLA in May 1987)
- Mr. Tom Y. Galloway (currently working in Geneva)
- Dr. Eduard H. Hovy (currently project member; attained Ph.D. in Computer Science from Yale University in May 1987)
- Dr. Robert T. Kasper (currently project member; attained Ph.D. in Computer Science from the University of Michigan in December 1986)
- Ms. Lynn Poulton (currently a Linguistics Department graduate student at the University of Sydney)
- Dr. William C. Mann (currently project member)

- Mr. Christian M.I.M. Matthiessen (currently a Linguistics Department faculty member at the University of Sydney; attained Ph.D. in Linguistics from UCLA in December 1988)
- Dr. Norman K. Sondheimer (currently head of the AI division of GE Corporate Research)
- Mr. Richard A. Whitney (currently project member; attained M.S. from UCLA in May 1988)

5 Interactions and Meetings

The 4th International Workshop on Natural Language Generation was held in July, 1988, with USC as sponsor (along with AAAI, ACL and ACM), hosted by Drs. Mann, Paris and Swartout from ISI. Although no AFOSR funds were applied to workshop expenses, the project benefited from extensive interactions with leaders in this kind of work. After the workshop, most of the conferees visited ISI for 1 to 3 days, which included numerous demonstrations of Penman and other research systems.

Our past research has gained enormously from visiting workers who have had no formal status on the project. Several eminent and highly qualified people have visited for periods of weeks, without pay, relating their work and expertise to the ongoing research. Visitors who stayed for at least two weeks include, from The Federal Republic of Germany Drs. H-J. Novak, B. Nebel, E. Steiner, J. Schütz; from Britain Ms. J. Wright; from Yugoslavia Dr. M. Simunović. Other visitors included Drs. D. Rösner, G. Kempen, K. Sparck-Jones, D. Weber, K. Shimohara; Messrs. N. Reithinger and M. Elhadad; and Ms. C. DiMarco.

The Penman text generation system has recently been structured as a distributable system, and has been distributed to seven institutions to date:

- University of Toronto, Toronto: loaded immediately, planned use in the thesis work of at least one graduate student.
- University of Delaware, Newark: awaiting acquisition of newest version of operating system. Planned for thesis work of two graduate students.
- EUROTRA-D, Saarbrücken, West Germany: awaiting porting to Sun computer.
- University of Illinois, Champaign: no status report yet.
- Columbia University, New York City: no status report yet; the intended use is to generate texts from the output of Columbia's text planner.
- University of Alabama, Huntsville: loaded immediately, being explored.
- York University, Toronto: being ported to Vax computer.

The following institutions have requested or expressed preliminary interest in Penman, but have not yet completed the licensing agreement:

- University of California, Berkeley.
- New Mexico State University, Las Cruces.
- Carnegie-Mellon University, Pittsburgh.
- Sydney University, Sydney.
- University of Massachusetts, Amherst.

The following institutions have, or have expressed the desire to acquire, a paper (non-computer-based) copy of Penman's grammar:

- IBM Natural Language Center, Los Angeles: currently installing the grammar on their own systems.
- York University, Toronto (English Department).
- University of Wales, Cardiff (English Department).

In addition to the publications listed in the previous section, the following presentations were made about work sponsored under this contract:

1. Mann, W. and Matthiessen, C. Functions of Language in Two Frameworks. Presented at the 14th *International Systemics Congress*, Sydney, Australia, July, 1987.
2. Matthiessen, C. and Mann, W. Rhetorical Structure Theory and Systemic Approaches to Text Generation. Presented at the 14th *International Systemics Congress*, Sydney, Australia, July, 1987.
3. Bateman, J., Kasper R. and Matthiessen, C.M.I.M. Systemic Linguistics and Natural Language Processing: Case Studies in the Exchange. Invited workshop presentation at the 15th *International Systemics Congress*, East Lansing, MI. August, 1988.
4. Bateman, J. Dynamic Systemic Functional Grammar: A New Frontier. Presented at the 15th *International Systemics Congress*, East Lansing, MI, August, 1988.
5. Kasper, R. Ambiguity in Systemic Grammar: Experience with a Computational Parser for English. Presented at the 15th *International Systemics Congress*, East Lansing, MI, August 1988.
6. Mann, W. Two Approaches to Discourse Structure from Computational Linguistics. Presented at the 15th *International Systemics Congress*, East Lansing, MI, August 1988.

7. Matthiessen, C. Notes on the Organization of the Environment of a Text Generation Grammar. In *Natural Language Generation: New results in Artificial Intelligence, Psychology and Linguistics*, Kempen, G. (ed.) Nijhoff, 1987. Also available as ISI report ISI/RS-86-177.

6 New Discoveries and Inventions

No new inventions or patent disclosures resulted from this work.

7 Other Statements Assisting Evaluation

No other statements are required to provide additional insight and information for an assessment of the work done under this contract.

8 References

1. [Cohen & Levesque 85] Cohen, P.R. & Levesque, H.J. Speech Acts and Rationality. In *Proceedings of the ACL Conference*, Chicago, 1985.

2. [Grosz & Sidner 86] Grosz, B.J. & Sidner, C.L. Attention, Intentions, and the Structure of Discourse. In *Computational Linguistics Journal* 12(3), 1986.

3. [Hovy 88a] Hovy, E.H. Planning Coherent Multisentential Text. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June, 1988. Also available as USC/Information Sciences Institute Reprint RS-88-208.

4. [Hovy 88b] Hovy, E.H. Approaches to the Planning of Coherent Text. In *Proceedings of the 4th International Generation Workshop*, Los Angeles, CA, July, 1988.

5. [Kasper 87a] Kasper, R.T. A Unification Method for Disjunctive Feature Descriptions. In *Proceedings of the* 25th *Annual Meeting of the Association for Computational Linguistics*, Palo Alto, CA, July, 1987. Also available as USC/Information Sciences Institute Reprint RS-87-187.

6. [Kasper 87b] Kasper, R.T. Conditional Descriptions in Functional Unification Grammar. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June 1988. Also available as USC/Information Sciences Institute Research Report RR-87-191, November, 1987.

7. [Kasper 88] Kasper, R.T. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

Hungary, August 1988. Also available as USC/Information Sciences Institute Reprint RS-88-212.

8. [Mann 88a] Mann, W.C. Text Generation: The Problem of Text Structure. In *Natural Language Generation Systems*, McDonald, D. and Bolc, L. (eds), Springer-Verlag: New York, 1988.

9. [Mann 88b] Mann, W.C. Dialogue Games. In *Argumentation*, 1988. Also available as USC/Information Sciences Institute report RR-79-77.

10. [Mann 88c] Mann, W.C. Two Theories of Discourse Structure. In *Proceedings of the 4th International Generation Workshop*, Los Angeles, CA, July, 1988.

11. [Mann, Matthiessen & Thompson 88] Mann, W.C., Matthiessen, C.M.I.M. & Thompson, S.A. Rhetorical Structure Theory and Text Analysis. In *Discourse Description: Diverse Analyses of a Fund Raising Text*, Mann, W. and Thompson, S. (eds), (to appear).

12. [Mann & Thompson 83] Mann, W.C. & Thompson, S.A. Relational Propositions in Discourse. USC/Information Sciences Institute Research Report RR-83-115, 1983.

13. [Mann & Thompson 87a] Mann, W.C. & Thompson, S.A. Rhetorical Structure Theory: A Framework for the Analysis of Texts. In *IPRA Papers in Pragmatics*, Volume 1, 1987. Also available as USC/Information Sciences Institute report RS-87-185.

14. [Mann & Thompson 87b] Mann, W.C. & Thompson, S.A. Rhetorical Structure Theory: Description and Construction of Text Structures. In *Natural Language Generation: New results in Artificial Intelligence, Psychology and Linguistics*, Kempen, G. (ed.), Nijhoff, 1987. Also available as USC/Information Sciences Institute report RS-87-174.

15. [Mann & Thompson 88a] Mann, W.C. & Thompson, S.A. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. In *Text*, Vol. 8:3, 1988.

16. [Mann & Thompson 88b] Mann, W.C. & Thompson, S.A. Rhetorical Structure Theory: A Theory of Text Organization. In *The Structure of Discourse*, Polanyi, L. (ed), Ablex: Norwood, NJ, 1988.

17. [McKeown 82] McKeown, K.R. *Generating Natural Language Text in Response to Questions about Database Queries*. Ph.D. dissertation, University of Pennsylvania, 1982.

18. [Sacerdoti 77] Sacerdoti, E. *A Structure for Plans and Behavior*. North-Holland, 1977.

19. [Thompson & Mann 86] Thompson, S.A. & Mann, W.C. A Discourse View of Concession in Written English. In *Proceedings of the Second Annual Meeting of the Pacific Linguistics Conference*, DeLancey, S, and Tomlin, R., 1986.

20. [Thompson & Mann 87] Thompson, S.A. & Mann, W.C. Antithesis: A Study in Clause Combining and Discourse Structure. In *Language Topics: Essays in Honour of M.A.K. Halliday*, Steele, R. and Threadgold, T. (eds.), Benjamins, 1987. Also available as USC/Information Sciences Institute report RS-87-171.